

# Two Dichotomy Theorems

Domenico Cantone

*Dipartimento di Matematica e Informatica, Università di Catania  
Viale Andrea Doria 6, I-95125 Catania, Italy.  
E-mail: cantone@dmf.unict.it*

Pietro Ursino

*Dipartimento di Fisica e Matematica, Università dell'Insubria  
Via Valleggio 11, I-22100 Como, Italy.  
E-mail: pietro.ursino@uninsubria.it*

## Introduction

The core argument of Computable Set Theory is the problem of decidability of some fragments of Set Theory. The general features of this area of research have been widely settled in at least two books [CFO89] and [COP01]. A quite standard argument in order to set decidability of a language consists in proving a small model property that is to prove that whenever a formula of the language has model it has a model of a rank not exceeding a fixed number which depends only on the number of variables involved in the assigned formula.

In the following we deal with a particular behaviour of some set theoretic languages: the dichotomy property. This property says, roughly speaking, that whenever a formula of a fixed language is satisfiable it admits a model smaller than a fixed rank (depending only on the number of the variables of the formula) otherwise it admits an infinite model. This property depends strongly on the expressivity of the language indeed the more a language has expressive power the more it is hard to close its models in a cage, which is the key to prove the decidability. Dichotomy is, in a sense, a medium point between small model property and undecidability since it does not assert that the model has a cage, but says that when this does not happen the model can diverge until the infinite. What it is forbidden is that admits unbound finite models without admitting an infinite model.

In the following we give a formal definition of the above discussed dichotomy property. Let  $\mathfrak{F}$  be a fragment of set theory. For a set assignment  $M$  and a formula  $\varphi \in \mathfrak{F}$ , we put

$$\text{dom}_\varphi(M) := \bigcup \{Mx \mid x \in \text{Vars}(\varphi)\}.$$

We say that a formula  $\varphi \in \mathfrak{F}$  admits only finite models if for every set assignment  $M$  we have

$$M \models \varphi \quad \implies \quad |\text{dom}_\varphi(M)| < \omega.$$

**Definition 0.1.** A fragment  $\mathfrak{F}$  of set theory is dichotomic if there exists a map  $f: \omega \rightarrow \omega$  (called DICHOTOMIC MAP FOR  $\mathfrak{F}$ ) such that for every  $\varphi \in \mathfrak{F}$  admitting only finite models

and for every set assignment  $M$  we have

$$M \models \varphi \quad \implies \quad \text{rk}(\text{dom}_\varphi(M)) < f(|\text{Vars}(\varphi)|).$$

The main results showed in the present paper are the following theorems

**Theorem 0.2.** *The theory MLSP is dichotomic.*

**Theorem 0.3.** *The theory MLSCNOTORD is not dichotomic.*

and using this last result:

**Theorem 0.4.** *MLSCNOTORD with disjoint unary union is undecidable*

In order to deal with those kinds of problems, we largely make use of the theory of formative processes (see [Urs05] , [CU14] and [CU17]). In particular we focus on the way in which it manages with the cycles of a P-graph, which can behave in a different way for different languages. Cycles are commonly seen as generators of infinite elements and they could be bounded in this production only by an external constraint, for example if one of the regions involved in the cycle is a component of an assignment  $Mx$  of a variable  $x$  which appears in the formula with literals as  $Finite(x)$  or  $x = \{y\}$ . In this case, since the cycle has a cardinal bond, it cannot be pumped. In case of MLSC or MLSCNOTORD we are in presence of an internal bond as it can be seen in the proof of Theorem 0.3. Indeed, still there is an infinite production of elements but they are just necessary to keep the "engine running". This phenomenon allows the creation of models of increasing rank never reaching the infinite, which in turns implies MLSCNOTORD is not dichotomic.

## 1 Syntax and semantics of the theory $\mathcal{S}$

The syntax of the quantifier-free fragment  $\mathcal{S}$  is defined as follows<sup>1</sup>. The symbols of  $\mathcal{S}$  are:

- infinitely many set variables  $x, y, z, \dots$ ;
- the constant symbol  $\emptyset$ ;
- the set operators  $\cdot \cup \cdot, \cdot \cap \cdot, \cdot \setminus \cdot, \cdot \times \cdot, \cdot \otimes \cdot, \{\cdot, \dots, \cdot\}, \mathcal{P}(\cdot), \bigcup(\cdot), \bigcap(\cdot)$ ;
- the set predicates  $\cdot \subseteq \cdot, \cdot = \cdot, \cdot \in \cdot, Finite(\cdot)$ .

The set of  $\mathcal{S}$ -TERMS is the smallest collection of expressions such that:

- all variables and the constant  $\emptyset$  are  $\mathcal{S}$ -terms;
- if  $s$  and  $t$  are  $\mathcal{S}$ -terms, so are  $s \cup t, s \cap t, s \setminus t, s \times t, s \otimes t, \mathcal{P}(s), \bigcup s$ , and  $\bigcap s$ ;
- if  $s_1, \dots, s_n$  are  $\mathcal{S}$ -terms, so is  $\{s_1, \dots, s_n\}$ .

$\mathcal{S}$ -ATOMS have then the form

$$s \subseteq t, \quad s = t, \quad s \in t, \quad Finite(s),$$

---

<sup>1</sup>By  $\otimes$  we denote the unordered cartesian product.

where  $s, t$  are  $\mathcal{S}$ -terms.

$\mathcal{S}$ -FORMULAE are propositional combinations of  $\mathcal{S}$ -atoms, by means of the usual logical connectives  $\wedge$  (conjunction),  $\vee$  (disjunction),  $\neg$  (negation),  $\rightarrow$  (implication),  $\leftrightarrow$  (bi-implication), etc.  $\mathcal{S}$ -LITERALS are  $\mathcal{S}$ -atoms and their negations.

For a  $\mathcal{S}$ -formula  $\Phi$ , we denote by  $\text{Vars}(\Phi)$  the collection of set variables occurring in  $\Phi$  (similarly for  $\mathcal{S}$ -terms).

$\{x \cup \emptyset, \mathcal{P}(y \cap (z \setminus \{x\}))\} \setminus \bigcup(x \cap y)$	$\mathcal{S}$ -term
$\neg(z \cup x \in \{x \cup \emptyset, \mathcal{P}(y \cap (z \setminus \{x\}))\} \rightarrow (x \notin \bigcap z \vee z \in \bigcup x))$	$\mathcal{S}$ -formula
$\text{Vars}\left(\neg(z \cup x \in \{x \cup \emptyset, \mathcal{P}(y \cap (z \setminus \{x\}))\} \rightarrow (x \notin \bigcap z \vee z \in \bigcup x))\right) = \{x, y, z\}$	

Table 1: Some examples: an  $\mathcal{S}$ -term, an  $\mathcal{S}$ -formula, and the map  $\text{Vars}(\cdot)$ .

Our considerations will take place in a naive set theory which could be formalizable in the standard axiom system  $ZFC$ , developed by Zermelo, Fraenkel, Skolem, and von Neumann (see [Jec78]). In particular, they will refer to the von Neumann standard cumulative hierarchy of sets, a very specific model of  $ZFC$ , and will assume the AXIOM OF REGULARITY. Then semantics of  $\mathcal{S}$  is defined in the obvious way. A SET ASSIGNMENT  $M$  is any map from a collection  $V$  of set variables (called the DOMAIN OF  $M$ ) into the universe  $\mathfrak{V}$  of all sets (in short,  $M \in \mathfrak{V}^V$  or  $M \in \{\text{sets}\}^V$ ). Given set assignment  $M$  over a collection of variables  $V$ , the SET DOMAIN OF  $M$  is the set  $\bigcup M[V] = \bigcup_{v \in V} Mv$  and the RANK OF  $M$  is the ordinal

$$\text{rk}(M) := \text{rk}(\bigcup M[V])$$

(so that, when  $V$  is finite,  $\text{rk}(M) = \max_{v \in V} \text{rk}(Mv)$ ). A set assignment  $M$  is FINITE, if so is its set domain.

Let  $M$  be a set assignment over a given collection  $V$  of variables, and let  $s, t, s_1, \dots, s_n$  be  $\mathcal{S}$ -terms whose variables occur in  $V$ . We put, recursively,

$$\begin{aligned}
M\emptyset &:= \emptyset \\
M(s \cup t) &:= Ms \cup Mt \\
M(s \cap t) &:= Ms \cap Mt \\
M(s \setminus t) &:= Ms \setminus Mt \\
M(s \times t) &:= Ms \times Mt \\
M(s \otimes t) &:= Ms \otimes Mt \\
M(\mathcal{P}(s)) &:= \mathcal{P}(Ms) := \{u \mid u \subseteq Ms\} \\
M\left(\bigcup s\right) &:= \bigcup Ms := \{u \mid u \in u' \in Ms, \text{ for some } u'\} \\
M\left(\bigcap s\right) &:= \bigcap Ms := \{u \mid u \in u', \text{ for every } u' \in Ms\} \quad (\text{provided that } Ms \neq \emptyset).
\end{aligned}$$

We also put

$$\begin{aligned} (s \in t)^M &:= \begin{cases} \mathbf{t} & \text{if } Ms \in Mt \\ \mathbf{f} & \text{otherwise} \end{cases} & (s = t)^M &:= \begin{cases} \mathbf{t} & \text{if } Ms = Mt \\ \mathbf{f} & \text{otherwise} \end{cases} \\ (s \subseteq t)^M &:= \begin{cases} \mathbf{t} & \text{if } Ms \subseteq Mt \\ \mathbf{f} & \text{otherwise} \end{cases} & (Finite(s))^M &:= \begin{cases} \mathbf{t} & \text{if } Ms \text{ is finite} \\ \mathbf{f} & \text{otherwise} \end{cases} \end{aligned}$$

(where, plainly,  $\mathbf{t}$  and  $\mathbf{f}$  stand the truth-values *true* and *false*, respectively), and

$$\begin{aligned} (\Phi \wedge \Psi)^M &:= \Phi^M \wedge \Psi^M & (\Phi \vee \Psi)^M &:= \Phi^M \vee \Psi^M \\ (\Phi \rightarrow \Psi)^M &:= \Phi^M \rightarrow \Psi^M & (\neg\Phi)^M &:= \neg(\Phi^M) \quad \text{etc.} \end{aligned}$$

for all  $\mathcal{S}$ -formulae  $\Phi, \Psi$  such that  $\text{Vars}(\Phi), \text{Vars}(\Psi) \subseteq V$ .

The set assignment  $M$  is said to **SATISFY** an  $\mathcal{S}$ -formula  $\Phi$  if  $\Phi^M = \mathbf{t}$  holds, in which case we also write  $M \models \Phi$  and say that  $M$  is a **MODEL** for  $\Phi$ . If  $\Phi$  has a model, we say that  $\Phi$  is **SATISFIABLE**; otherwise, we say that  $\Phi$  is **UNSATISFIABLE**. If  $\Phi$  has a finite model, we say that it is **FINITELY SATISFIABLE**. If  $\Phi$  has a model  $M$  such that  $Mx \neq My$  for all distinct variables  $x, y \in \text{Vars}(\Phi)$ , we say that it is **INJECTIVELY SATISFIABLE**. If  $M' \models \Phi$  for every set assignment  $M'$  defined over  $\text{Vars}(\Phi)$ , then  $\Phi$  is said to be **TRUE**. Two  $\mathcal{S}$ -formulae  $\Phi$  and  $\Psi$  are said to be **EQUISATISFIABLE** if  $\Phi$  is satisfiable if and only if so is  $\Psi$ .

### 1.1 The decision problem for subtheories of $\mathcal{S}$

Let  $\mathcal{S}'$  be any subtheory of  $\mathcal{S}$ . The **DECISION PROBLEM** (or **SATISFIABILITY PROBLEM**, or **SATISFACTION PROBLEM**) for  $\mathcal{S}'$  is the problem of establishing algorithmically whether any given  $\mathcal{S}'$ -formula is satisfiable. If the decision problem for  $\mathcal{S}'$  is solvable, then  $\mathcal{S}'$  is said to be **DECIDABLE**. A **DECISION PROCEDURE** (or **SATISFIABILITY TEST**) for  $\mathcal{S}'$  is any algorithm which solves the decision problem for  $\mathcal{S}'$ . The **FINITE SATISFIABILITY PROBLEM** for  $\mathcal{S}'$  is the problem of establishing algorithmically whether any given  $\mathcal{S}'$ -formula is finitely satisfiable. The **INJECTIVE SATISFIABILITY PROBLEM** for  $\mathcal{S}'$  is the problem of establishing algorithmically whether any given  $\mathcal{S}'$ -formula is injectively satisfiable.

By making use of the disjunctive normal form, the satisfiability problem for  $\mathcal{S}$  can be readily reduced to the same problem for conjunctions of  $\mathcal{S}$ -literals. In addition, by suitably introducing fresh set variables to name subterms of the following types

$$t_1 \cup t_2, \quad t_1 \cap t_2, \quad t_1 \setminus t_2, \quad t_1 \times t_2, \quad t_1 \otimes t_2, \quad \mathcal{P}(t), \quad \bigcup t, \quad \bigcap t,$$

where  $t_1, t_2, t$  are  $\mathcal{S}$ -terms, the satisfiability problem for  $\mathcal{S}$  can further be reduced to the satisfiability problem for conjunctions of  $\mathcal{S}$ -literals of the following types

$$\begin{aligned} x = y \cup z, & \quad x = y \cap z, & x = y \setminus z, & \quad x = \{y_1, \dots, y_H\}, & x = \mathcal{P}(y), \\ x = \bigcup y, & \quad x = \bigcap y, & x = y, & \quad x \neq y, & s \times t, \\ s \otimes t, & \quad x \in y, & x \notin y, & \quad x \subseteq y, & x \not\subseteq y, \end{aligned} \tag{1}$$

where  $x, y, z, y_1, \dots, y_H$  stand for set variables or the constant  $\emptyset$ .

Finally, by applying the following simplification rules

(s1) a literal of type  $x = y$  can be replaced by the equivalent literal  $x = y \cup y$ ,

- (s2) a literal of type  $x \not\subseteq y$  is equisatisfiable with  $z' = x \setminus y \wedge z' \neq \emptyset$ ,
- (s3) the constant  $\emptyset$  can be eliminated by replacing it with a new variable  $y_\emptyset$  and adding the conjunct  $y_\emptyset = y_\emptyset \setminus y_\emptyset$ ,
- (s4) a literal of type  $x = y \cap z$  is equisatisfiable with  $y' = y \setminus z \wedge x = y \setminus y'$ ,
- (s5) a literal of type  $x \subseteq y$  is equisatisfiable with  $y = x \cup y$ ,
- (s6) a literal of type  $x \neq y$  is equisatisfiable with  $x \in z' \wedge y \notin z'$ ,
- (s7) a literal of type  $x \notin y$  is equisatisfiable with  $x \in z' \wedge z' = z' \setminus y$ ,

(where  $y'$  and  $z'$  stand for fresh set variables), the satisfiability problem for  $\mathcal{S}$  can be reduced to the satisfiability problem for conjunctions of  $\mathcal{S}$ -atoms of the following types:

$$\begin{array}{llllll} x = y \cup z, & x = y \setminus z, & x \in y, & x = \{y_1, \dots, y_H\}, & s \otimes t & \\ s \times t, & x = \bigcup y, & x = \bigcap y, & x = \mathcal{P}(y) & & \end{array} \quad (2)$$

(where  $x, y, y_1, \dots, y_H$  stand for variables), which we call **NORMALIZED CONJUNCTIONS OF  $\mathcal{S}$** . Needless to say, working with normalized conjunctions simplifies the completeness and correctness proofs of decision procedures.

The reduction technique for the satisfiability problem to normalized conjunctions has been illustrated in the case of the whole theory  $\mathcal{S}$ ; however, it can directly be adapted to subtheories of  $\mathcal{S}$ .

Notice that a satisfiability test for a subtheory  $\mathcal{S}'$  of  $\mathcal{S}$  can also be used to decide whether any given formula  $\Phi$  in  $\mathcal{S}'$  is **TRUE**. In fact, a formula  $\Phi$  is true if and only if its negation  $\neg\Phi$  is unsatisfiable.

### 1.1.1 Decidable fragments of set theory

Over the years, several subtheories of  $\mathcal{S}$  have been proved to be decidable. We mention in particular the theory Multi-Level Syllogistic (MLS), which is the common kernel of most decidable fragments of set theory investigated in the field of Computable Set Theory. Specifically, **MLS** is the propositional combination of atomic formulae of the following three types<sup>2</sup>

$$x = y \cup z, \quad x = y \setminus z, \quad x \in y. \quad (3)$$

Below we give the list of the subtheories of  $\mathcal{S}$  which we deal with in the present article:

MLS:	$\cup, \cap, \setminus, \subseteq, =, \in$	(cf. [FOS80])
MLSP:	$\cup, \cap, \setminus, \subseteq, =, \in, \mathcal{P}(\cdot)$	(cf. [CFS85])
MLSC:	$\cup, \cap, \setminus, \subseteq, =, \in, \times$	(Open problem)
MLSCNOTORD:	$\cup, \cap, \setminus, \subseteq, =, \in, \otimes$	(Open problem)
MLSSPF:	$\cup, \cap, \setminus, \subseteq, =, \in, \{\cdot\}, \mathcal{P}(\cdot), Finite(\cdot)$	(cf. [CU14])
...	...	...

<sup>2</sup>As remarked above, intersection and set inclusion are easily expressible by means of literals of type (3).

The interested reader can find an extensive treatment of such results in [CFO89] and [COP01].

In several cases, the decidability of a given fragment of set theory has been shown by proving for it a small model property.

## 2 The theory MLSP

We proceed by describing some examples of the expressive power of MLSP, then we prove that it is dichotomic.

### 2.1 Expressing hereditarily finite sets with MLSP-conjunctions

We show that it is possible to express hereditarily finite sets in the theory MLSP, though the singleton operator is not a primitive operator of MLSP.

For every set  $s$ , we plainly have

$$\mathcal{P}(s) = \bigcup_{s'' \subsetneq s} \{s''\},$$

where the singletons on the left-and-side are all pairwise disjoint. Hence, in particular,

$$\begin{aligned} \{s\} &= \mathcal{P}(s) \setminus \bigcup_{s'' \subsetneq s} \{s''\} \\ &= \mathcal{P}\left(\bigcup_{s' \in s} \{s'\}\right) \setminus \bigcup_{s'' \subsetneq s} \{s''\} \end{aligned} \tag{4}$$

From (4) it follows that the singleton of a finite set  $s$  can be expressed by a finite expression involving the singletons of the elements and of the proper subsets of  $s$ , and the operators of binary union, set difference, and powerset.

The above observation suggests a way to represent each hereditarily finite set with a suitable formula of the theory MLSP involving only equality ( $=$ ), conjunction ( $\wedge$ ), the standard Boolean set operators  $\cup$  and  $\setminus$ , and the powerset operator  $\mathcal{P}$ ; thus, neither the membership relator nor the singleton operator are used. We recall that *hereditarily finite* sets are those sets that are *finite* and whose elements, elements of elements, etc., all are finite. We denote their collection by HF. Plainly,  $\text{HF} = \mathcal{V}_\omega$ , where  $\omega = \{0, 1, 2, \dots\}$  is the first infinite ordinal.<sup>3</sup> Observe also that  $\mathcal{V}_n \in \text{HF}$ , for every  $n \in \omega$ .

In our representation, we will use only variables of the form  $x_{\{h\}}$ , indexed by singletons of hereditarily finite sets. In addition, it will turn out that each representing formula  $\varphi$  is satisfiable and also enjoys the following *faithfulness condition*:

- if  $M \models \varphi$ , then for each variable  $x_{\{h\}}$  occurring in  $\varphi$  it must be the case that  $M x_{\{h\}} = \{h\}$ , where  $h \in \text{HF}$ .

---

<sup>3</sup>We are using here the well-known von Neumann encoding of integers, recursively defined as  $0 := \emptyset$ , and  $i + 1 := i \cup \{i\}$ .

For each  $h \in \mathbf{HF}$ , we recursively define the representing MLSP-formula  $\varphi_{\{h\}}$  by putting

$$\varphi_{\{h\}} := \begin{cases} (x_{\{\emptyset\}} = \mathcal{P}(x_{\{\emptyset\}} \setminus x_{\{\emptyset\}})) & \text{if } h = \emptyset \\ \left( x_{\{h\}} = \mathcal{P}\left(\bigcup_{h' \in h} x_{\{h'\}}\right) \setminus \bigcup_{h'' \subsetneq h} x_{\{h''\}} \right) \wedge \bigwedge_{h' \in h} \varphi_{\{h'\}} \wedge \bigwedge_{h'' \subsetneq h} \varphi_{\{h''\}} & \text{if } h \neq \emptyset. \end{cases} \quad (5)$$

To show that the recursive definition (5) is well-given, one has to exhibit a well-ordering  $\prec$  of  $\mathbf{HF}$  such that, for  $h \in \mathbf{HF}$ , the following properties hold:

(P1)  $h' \prec h$ , for every  $h' \in h$ , and

(P2)  $h'' \prec h$ , for every  $h'' \subsetneq h$ .

In particular any total ordering  $\prec$  of  $\mathbf{HF}$  complying with the rank, i.e., such that

$$\text{rk } h' < \text{rk } h \implies h' \prec h,$$

and extending the strict partial ordering  $\subsetneq$  among sets of the same rank, i.e., such that

$$[(\text{rk } h' = \text{rk } h) \wedge (h' \subsetneq h)] \implies h' \prec h,$$

satisfies (P1) and (P2) above. This is the case, for instance, for the ordering on  $\mathbf{HF}$  induced by the Ackermann encoding

$$\mathbb{N}(h) = \sum_{h' \in h} 2^{\mathbb{N}(h')},$$

for  $h \in \mathbf{HF}$ . Indeed, it is an easy matter to check that for each  $h \in \mathbf{HF}$  we have

- $\mathbb{N}(h') < \mathbb{N}(h)$ , for every  $h' \in h$ , and
- $\mathbb{N}(h'') < \mathbb{N}(h)$ , for every  $h'' \subsetneq h$ .

By induction on the Ackermann code  $\mathbb{N}(h)$ , it can easily be shown that each MLSP-formula  $\varphi_{\{h\}}$  is satisfiable and it also satisfies the above faithfulness condition. Hence, for every hereditarily finite set  $h$ , the formula

$$x_h \in x_{\{h\}} \wedge \varphi_{\{h\}}$$

faithfully expresses  $h$  via the variable  $x_h$ .

Alternatively, a hereditarily finite set  $h$  can also be expressed in MLSP, via the variable  $x_h$ , as follows:

$$\begin{cases} x_{\emptyset} = x_{\emptyset} \setminus x_{\emptyset} & \text{if } h = \emptyset \\ \left( x_h = \bigcup_{h' \in h} x_{\{h'\}} \right) \wedge \bigwedge_{h' \in h} \varphi_{\{h'\}} & \text{otherwise.} \end{cases}$$

Finally, we observe that each MLSP-formula  $\varphi_{\{h\}}$  can be easily transformed into an equisatisfiable MLSP-conjunction  $\varphi'_{\{h\}}$  involving newly introduced variables.

## 2.2 Expressing the operator $\mathcal{P}^*(s_1, \dots, s_k)$ with MLSP-conjunctions

Given a (possibly empty) finite list of sets  $s_1, \dots, s_k \in \mathcal{V}$ ,  $\mathcal{P}^*(s_1, \dots, s_k)$  is the collection of the subsets of  $s_1 \cup \dots \cup s_k$  which have non-null intersection with each of the sets  $s_1, \dots, s_k$ ; in symbols,

$$\mathcal{P}^*(s_1, \dots, s_k) := \left\{ s \subseteq \bigcup_{i=1}^k s_i \mid s \cap s_i \neq \emptyset, \text{ for } i = 1, \dots, k \right\}. \quad (6)$$

Thus, in particular, for the empty list  $\Lambda$ , we have  $\mathcal{P}^*(\Lambda) = \{\emptyset\}$ . In addition, we have

$$\mathcal{P}^*(s_1, \dots, s_k) = \emptyset \iff \bigvee_{i=1}^k s_i = \emptyset. \quad (7)$$

and

$$\mathcal{P}^*(s_1, s_2) = \mathcal{P}(s_1 \cup s_2) \setminus (\mathcal{P}(s_1 \setminus s_2) \cup \mathcal{P}(s_2 \setminus s_1)). \quad (8)$$

The equivalence (7) follows immediately from the very definition of  $\mathcal{P}^*(\{s_1, \dots, s_k\})$ . Concerning (8), if  $s \in \mathcal{P}^*(s_1, s_2)$ , then plainly  $s \subseteq s_1 \cup s_2$ ,  $s \not\subseteq s_1 \setminus s_2$ , and  $s \not\subseteq s_2 \setminus s_1$ , so that  $s \in \mathcal{P}(s_1 \cup s_2) \setminus (\mathcal{P}(s_1 \setminus s_2) \cup \mathcal{P}(s_2 \setminus s_1))$ . Conversely, if  $s \in \mathcal{P}(s_1 \cup s_2) \setminus (\mathcal{P}(s_1 \setminus s_2) \cup \mathcal{P}(s_2 \setminus s_1))$ , then  $s \subseteq s_1 \cup s_2$  and  $s \cap s_1 \neq \emptyset \neq s \cap s_2$ , which yields, by (6),  $s \in \mathcal{P}^*(s_1, s_2)$ . Hence, (8) follows.

Equation (8) readily generalizes to

$$\mathcal{P}^*(s_1, \dots, s_k) = \mathcal{P}(\mathcal{S}) \setminus \left( \bigcup_{i=1}^k \mathcal{P}(\mathcal{S} \setminus s_i) \right), \quad (9)$$

where  $\mathcal{S} := \bigcup_{i=1}^k s_i$ .

From (9) it follows immediately that a literal of the form  $x = \mathcal{P}^*(y_1, \dots, y_k)$  can be expressed by the MLSP-literal

$$x = \mathcal{P}\left(\bigcup_{i=1}^k y_i\right) \setminus \left(\bigcup_{j=1}^k \mathcal{P}\left(\left(\bigcup_{i=1}^k y_i\right) \setminus y_j\right)\right),$$

which, as before, can in turn be transformed into an equisatisfiable MLSP-conjunction involving newly introduced variables.

## 2.3 The subtheory $\widehat{\text{MLSP}}$

We investigate the subtheory of MLSP, denoted  $\widehat{\text{MLSP}}$ , consisting of the collection of MLSP-formulae which do not admit infinite models.

We provide the following semantic definition for  $\widehat{\text{MLSP}}$ .

**Definition 2.1.**  $\widehat{\text{MLSP}} := \{\psi \in \text{MLSP} \mid \psi \wedge (\bigvee_{x \in \text{Vars}(\psi)} \neg \text{Finite}(x)) \text{ is unsatisfiable}\}.$

□

At a first glance it seems that arguing from the fact that MLSP has a decidable satisfiability problem (cf. [CFS85]) we can assert that the satisfiability problem for  $\widehat{\text{MLSP}}$  is



decidable, since  $\widehat{\text{MLSP}} \subseteq \text{MLSP}$ . Unfortunately the problem of deciding if a formula  $\psi$  is or is not in  $\widehat{\text{MLSP}}$  is not a trivial one therefore using the decision procedure of **MLSP** you could find out that it is, for example, satisfiable but the procedure do not tell you if  $\psi$  is inside  $\widehat{\text{MLSP}}$  or not. We summarize the preceding remark by asserting that the real decision procedure for  $\widehat{\text{MLSP}}$  is advocated by the decision procedure for **MLSSPF** (namely, **MLSSP** with the finiteness predicate  $\text{Finite}(\cdot)$ ; cf. [CU14]).

Let  $\psi$  be an **MLSP**-formula. We say that  $\psi$  satisfies the *ordinal condition* if

$$(\star) \quad \{\text{rk}(Mx) \mid x \in \text{Vars}(\psi)\} \text{ is an ordinal, for every model } M \text{ of } \psi.$$

Observe that when  $\{\text{rk}(Mx) \mid x \in \text{Vars}(\psi)\}$  is an ordinal, then  $\{\text{rk}(Mx) \mid x \in \text{Vars}(\psi)\} = \text{rk}(M)$ , and conversely.

If the ordinal condition holds for  $\psi$ , then  $\psi \in \widehat{\text{MLSP}}$ , since then  $\text{rk}(M) < |\text{Vars}(\psi)|$  holds for every model  $M$  of  $\psi$ , and therefore

$$\psi \wedge (\bigvee_{x \in \text{Vars}(\psi)} \neg \text{Finite}(x))$$

is unsatisfiable.

We claim that the ordinal condition  $(\star)$  is also necessary for  $\psi \in \widehat{\text{MLSP}}$  to hold, i.e.:

**Conjecture 1.** *If  $\psi \in \widehat{\text{MLSP}}$ , then  $\{\text{rk}(Mx) \mid x \in \text{Vars}(\psi)\}$  is an ordinal, for every model  $M$  of  $\psi$ .*

### 3 MLSP is dichotomic

The following theorem proves Conjecture 1.

**Theorem 3.1.** *Let  $\varphi \in \text{MLSP}$  a satisfiable formula,  $M$  a model of  $\varphi$  with related transitive partition  $\Sigma_l$  with formative process  $\varpi = \langle (\Sigma_i)_{i \leq l}, (\bullet), \mathcal{T}, \mathcal{Q}_{\sigma^*} \rangle$  longer than  $k = 2^{2^{|\text{Var}(\varphi)|}}$ . Then there exists an infinite model of  $\varphi$ .*

*Proof.* It can easily be checked that  $\Sigma_l$  has  $P$ -graph with a cycle,  $\mathcal{C}$ , which the formative process follows at least two times. Consider the following set of variables of  $\varphi$ :

$$\mathcal{V}_{\mathcal{C}} = \{x \in \text{Var}(\varphi) \mid \exists \sigma \in \mathcal{C} \wedge \sigma \subseteq x\}$$

Fix  $x \in \mathcal{V}_{\mathcal{C}}$ . Define a new formula  $\varphi'$ :

$$\varphi' = \varphi \wedge \neg \text{Finite}(x)$$

Observe that  $\varphi' \in \text{MLSSPF}$ . In order to prove our thesis it is sufficient to show that  $\mathcal{C}$  is a pumping cycle for  $\sigma$  (see Section 5.4.2). First we need to change our formative process in a colored formative process. With this aim in mind we consider the following procedure:

```

procedure LocalTrash(formProcess, pumpCycle);
  -- formProcess is a  $\mathcal{P}$ -process  $\varpi = \langle (\Sigma_\mu)_{\mu \leq \xi}, (\bullet), \mathcal{T}, \mathcal{Q}_{\sigma^*} \rangle$  and pumpCycle is a
  --  $\mathcal{C} = \langle \sigma_0, A_0, \sigma_1, A_1 \cdots \sigma_m, A_m = A_0 \rangle$ .
   $LT := \mathcal{C}_{places}$ 
  Label "unchecked" all nodes
  loop (until there is a node unchecked) pick a node  $B$  unchecked such that  $B \cap LT \neq \emptyset$  do
     $i := LC(B)$ 
    pick  $\sigma$  such that  $\Delta^{(i)}(\sigma_{i+1}) \neq \emptyset$ 
    put  $\sigma$  inside  $LT$ 
    label  $B$  checked
  return  $LT$ ;
end procedure;

```

The procedure above takes as input a finite formative process together with a simple cycle of the  $\mathcal{P}$ -graph of the resulting transitive partition and returns a collection of places  $LT$ .

Consider now the collection  $LT$  of places and the following colored  $\mathcal{P}$ -process  $\varpi = \langle (\Sigma_i)_{i \leq l}, (\bullet), \mathcal{T}, \mathcal{R}, \mathcal{Q}_{\sigma^*} \rangle$  where  $\mathcal{R} = \mathcal{P} \setminus LT$ .

By a simple check of procedure LocalTrash we can deduce that  $LT = \mathcal{P}$  is a  $\mathcal{P}$ -closed set of places, i.e.,  $LT \subseteq \mathcal{P} \setminus \mathcal{R}$  and every  $\mathcal{P}$ -node  $B \in \mathcal{Q}_{\sigma^*}$  that intersects  $\mathcal{P}$  has a local trash in it. Indeed, pick a node  $B$  such that  $B \cap LT \neq \emptyset$  by procedure Local Trash there is a  $\sigma' \in LT$  which receives elements in the Last Call of  $B$ . We show that such a  $\sigma'$  is inside  $LT$  and it is a local trash for  $B$ . The former is clearly fulfilled arguing from the construction of procedure Local Trash, the latter depends on the fact that all the nodes distribute all their elements, therefore for all  $A$  such that  $\sigma \in A$   $LC(A) \geq LC(B)$ . Since the cycle is repeated two times for all  $B \in \mathcal{Q}_{\sigma^*}$  that intersects  $\mathcal{P}$   $GE(B) \geq i_0$ , where  $i_0$  is the step in which the cycle is finished to be done. We can summarize the above results just saying that  $\mathcal{PC} = \langle \mathcal{C}, i_0, \sigma, LT \rangle$  is a pumping chain for  $\sigma$ . ■

## 4 The theories MLSC and MLSCNOTORD

Regarding syntax and semantics of MLSC and MLSCNOTORD we refers to Section 1.1.1.

We proceed by describing some examples of expressive power of MLSC.

We recall that by  $\times$  we mean ordered cartesian product á la Kuratowski.

### 4.1 Expressing hereditarily finite sets with MLSC-conjunctions

We show that the singleton operator is expressible in the theory MLSC in such a way to preserve rank-boundedness. Besides we show that it is possible to express hereditarily finite sets, as well.

Consider the following conjuncts in the first column (in the second column we indicate

some deductions):

$$\begin{array}{ll}
 (\alpha) & x \in x' \in w \in z = y' \times y' \quad |x'| \leq 2 \\
 (\beta) & x' \in y' \in w \quad \begin{array}{l} x' = \{x\} \text{ and } y' = \{x, \{x\}\} \\ w = \{\{x\}, \{x, \{x\}\}\} = (x, \{x\}) \\ z = \{x, \{x\}\} \times \{x, \{x\}\} \end{array}
 \end{array}$$

Let  $\psi$  be a formula of MLS extended with the Cartesian product,  $x$  a variable occurring in  $\psi$ , and  $x', y', w, z$  variables not occurring in  $\psi$ , and consider the formula

$$\psi' := \psi \wedge \alpha \wedge \beta.$$

Then for each model  $M$  of  $\psi$ , there is a unique extension  $M'$  of  $M$  over the variables  $x', y', w, z$  such that  $M'$  satisfies  $\psi'$ . In such a model we have

$$\begin{aligned}
 M'x' &= \{Mx\} \\
 M'y' &= \{Mx, \{Mx\}\} \\
 M'w &= \{\{Mx\}, \{Mx, \{Mx\}\}\} \\
 M'z &= \{Mx, \{Mx\}\} \times \{Mx, \{Mx\}\}.
 \end{aligned}$$

Besides, if the model  $M$  is rank-bounded,  $M'$  is so, as well.

Thus, the singleton operator can be expressed in MLS plus Cartesian product in such a way to preserve rank-boundedness.

The following lemma summarizes the above discussion in a more concise way.

**Lemma 4.1.** *The singleton operator is expressible in MLSC.*

*Proof.* The conjunction of the following literals expresses the literal  $y = \{x\}$ :

$$x, x' \in y', \quad x' \neq x, \quad y' \in y' \times y', \quad x \in y \subsetneq y',$$

where  $x'$  and  $y'$  are fresh variables.<sup>4</sup> ■

Since singletons are expressible, and also the empty set is readily expressible by the MLS-literal  $x_\emptyset = x_\emptyset \setminus x_\emptyset$ , it follows immediately that every hereditarily finite set is expressible in MLS extended with the Cartesian product (by rank-bounded formulae).

## 4.2 MLSCNOTORD is not dichotomic

We recall that by  $\otimes$  we mean not ordered cartesian product.

**Theorem 4.2.** *Let  $Y, Z$  be sets satisfying the following two conditions*

$$\{\emptyset\} \otimes (\{\emptyset\} \cup Y) = Z, \quad Y \subsetneq Z.$$

*Then  $\text{rk}(Z) < \omega$  and  $|Z \setminus Y| = 1$ .*

---

<sup>4</sup>It is not hard to see that the literals  $x' \neq x$  and  $y \subsetneq y'$  are expressible by conjunctions in MLSC.

*Proof.* Let  $Y, Z$  be sets satisfying the conditions of the theorem.

Let us define recursively the sequence of sets  $\{a_n\}_{n \in \omega}$ , where

$$\begin{aligned} a_0 &:= \{\emptyset\} \\ a_{n+1} &:= \{\emptyset, a_n\}, \end{aligned}$$

and put  $A := \{a_n \mid n \in \omega\}$ .

To begin with, we show that  $Z \subseteq A$ . We proceed by contradiction. If  $Z \not\subseteq A$ , let  $z \in Z \setminus A$  of minimal rank. Since  $z \in \{\emptyset\} \otimes (\{\emptyset\} \cup Y)$ , then  $z = \{\emptyset, y\}$ , for some  $y \in \{\emptyset\} \cup Y$ . In fact  $y \in Y$ , since otherwise we would have  $z = \{\emptyset\} \in A$ . Since  $\text{rk}(y) < \text{rk}(z)$  and  $y \in Z$ , we have  $y \in A$ , so that  $y = a_k$  for some  $k \geq 0$ . But then we would have  $z = a_{k+1} \in A$ , a contradiction.

Next we show that, for every  $k \geq 0$ ,

$$a_k \in Z \quad \implies \quad a_\ell \in Y, \text{ for every } 0 \leq \ell < k. \quad (10)$$

We proceed again by contradiction, and let  $\bar{k} \in \omega$  be minimal such that (10) does not hold for  $a_{\bar{k}}$ . But then  $a_{\bar{k}} = \{\emptyset, a_{\bar{k}-1}\}$ , with  $a_{\bar{k}-1} \in Y$ . Hence, by the minimality of  $\bar{k}$  it follows that  $a_{k'} \in Y$ , for all  $1 \leq k' < \bar{k} - 1$ , so that (10) holds for  $a_{\bar{k}}$ , a contradiction.

From (10) it follows that  $|Z \setminus Y| = 1$ . Indeed, if  $a_{k_1}, a_{k_2} \in Z \setminus Y$ , with  $k_1 \neq k_2$ , then, by (10), it would follow that  $\min(a_{k_1}, a_{k_2}) \in Y$ , a contradiction.

Finally, let  $Z \setminus Y = \{a_{\bar{k}}\}$ . From what we have shown, it follows that

$$Z = \{a_i \mid 0 \leq i \leq \bar{k}\},$$

and therefore  $\text{rk}(Z) = \bar{k} + 2 < \omega$ , concluding the proof of the theorem. ■

**Corollary 4.3.** *The theory MLSCNOTORD is not dichotomic.*

*Proof.* By Lemma 4.1 we can consider the following MLSCNOTORD-formula

$$\varphi(y, z) := \{\emptyset\} \otimes (\{\emptyset\} \cup y) = z \wedge y \subsetneq z.$$

By Lemma 4.2 for every set assignment  $M$ , we have  $M \models \varphi(y, z)$  if and only if there exists a  $k \in \omega$  such that

$$\begin{aligned} My &= \{a_\ell \mid \ell < k\} \\ Mz &= \{a_\ell \mid \ell \leq k\}. \end{aligned}$$

Thus,  $\varphi(y, z)$  admits only finite models, yet we have

$$\sup\{\text{rk}(Mz) \mid M \models \varphi(y, z)\} = \omega.$$

■

## 5 MLSCNOTORD with disjoint unary union is undecidable

For a set  $S$ , we define the disjoint unary union  $\uplus S$  by putting

$$\uplus S := \{t \mid (\exists! s)(s \in S \wedge t \in s)\}.$$

We shall show that the collection  $\mathfrak{C}$  of conjunctions of literals of the following types

$$\begin{aligned} x \in y, & & x = y \cup z, & & x = y \setminus z \\ x = \uplus y, & & x = y \otimes z \end{aligned} \tag{11}$$

has an undecidable satisfiability problem.

In view of the results in [CCP], it is sufficient to show that we can express the following *positive* literals

$$|x| = |y|, \quad \text{Finite}(x)$$

with conjunctions in  $\mathfrak{C}$ . This will be done in Facts 1 and 2 below, respectively.

The following fact states that positive literals of type  $|x| = |y|$  are expressible with conjunctions in  $\mathfrak{C}$ .

**Fact 1.** *Let  $\varphi$  be any conjunction in  $\mathfrak{C}$ , possibly involving the variables  $x$  and  $y$ , and let  $y'$  and  $z$  be any variables distinct from  $x$  and  $y$  and not occurring in  $\varphi$ . Let  $C(x, y, y', z)$  be the conjunction of the following three literals*

$$y' = \{x\} \otimes y, \quad z \subseteq x \otimes y', \quad \uplus z = x \cup y'.$$

*Then the formulae  $\varphi \wedge |x| = |y|$  and  $\varphi \wedge C(x, y, y', z)$  are equisatisfiable.*

*Proof.* We observe that Lemma 4.1 holds for MLSCNOTORD, as well. The thesis follows from the semantics of the operator  $\uplus$  and the validity of the following implication

$$y' = \{x\} \otimes y \implies (x \cap y' = \emptyset \wedge |y| = |y'|).$$

■

The following fact states that positive literals of type  $\text{Finite}(x)$  are expressible by conjunctions in  $\mathfrak{C}$ .

**Fact 2.** *Let  $\varphi$  be any conjunction in  $\mathfrak{C}$ , possibly involving the variable  $x$ , and let  $w$ ,  $y$ , and  $z$  be any variables distinct from  $x$  and not occurring in  $\varphi$ . Let  $F(x, w, y, z)$  be the conjunction of the following three literals*

$$z = \{\emptyset\} \otimes (\{\emptyset\} \cup w), \quad w \subsetneq z, \quad |x| = |w|.$$

*Then the formulae  $\varphi \wedge \text{Finite}(x)$  and  $\varphi \wedge F(x, w, y, z)$  are equisatisfiable.*

*Proof.* The thesis follows directly from Theorem 1 in Section 4.2. ■

**Remark 5.1.** *Fact 1 could be restated by using the standard unary union operator  $\cup$  in place of the disjoint unary union operator  $\uplus$ , but adopting the multiset semantics, rather than the standard semantics of set theory.*

Yet another alternative approach to prove Fact 1 would be to express literals of type  $|x| = |y|$  by means of the standard unary union operator  $\cup$  and of the predicate  $\text{isPartition}(\Sigma)$  expressing that  $\Sigma$  is a partition, namely a collection of pairwise disjoint sets.

Finally, another alternative approach to reduce Tenth's Hilbert problem (for an extensive treatise of this problem see [Mat93]) to a fragment of set theory would be to extend MLSC with just one occurrence of the predicate  $\text{isPartitionOf}(\Sigma, x)$ , expressing that  $\Sigma$  is a partition of  $x$ . Indeed, it is not hard to see that MLSC with a single positive conjunct of type  $|x| = |y|$  allows one to encode any Diophantine equation.

## References

- [CCP] D. Cantone, V. Cutello, and A. Policriti. Set-theoretic reductions of Hilbert's tenth problem. In E. Börger, H. Kleine Büning, and M. M. Richter, editors, *Proceedings of 3rd Workshop Computer Science Logic - CSL '89 (Kaiserslautern 1989)*, volume 440, pages 65–75, Berlin, 1990. Springer-Verlag.
- [CFO89] D. Cantone, A. Ferro, and E.G. Omodeo. *Computable Set Theory*, vol. no.6 Oxford Science Publications of *International Series of Monographs on Computer Science*. Clarendon Press, Oxford, UK, 1989.
- [CFS85] D. Cantone, A. Ferro, and J. T. Schwartz. Decision procedures for elementary sublanguages of set theory. VI. Multilevel syllogistic extended by the powerset operator. *Comm. Pure Appl. Math.*, XXXVIII(1):549–571, 1985.
- [COP01] D. Cantone, E.G. Omodeo, and A. Policriti. *Set Theory for Computing - From decision procedures to declarative programming with sets*. Monographs in Computer Science. Springer-Verlag, New York, 2001.
- [CU14] Domenico Cantone and Pietro Ursino. Formative processes with applications to the decision problem in set theory: II. Powerset and singleton operators, finiteness predicate. *Inf. Comput.*, 237:215–242, 2014.
- [CU17] Domenico Cantone and Pietro Ursino. *An Introduction to the Formative Processes Technique in Set Theory*. Springer-Verlag, New York, To Appear.
- [FOS80] A. Ferro, E.G. Omodeo, and J.T. Schwartz. Decision procedures for elementary sublanguages of set theory. I: Multilevel syllogistic and some extensions. *Comm. Pure Appl. Math.*, 33:599–608, 1980.
- [Jec78] T.J. Jech. *Set Theory*. Academic Press, New York, 1978.
- [Mat93] Y.V. Matiassevitch. *Hilbert's Tenth Problem*. The MIT Press, Cambridge, MA, 1993.
- [Urs05] Pietro Ursino. A generalized small model property for languages which force the infinity. *Matematiche (Catania)*, LX(I):93–119, 2005.